Developer Guide

Chamo

By **Anonymous Messaging**:

Alex Kem

Angeline Dequit

Sopheak Chim

Darren Seng

Ivy Ly

Version: 3.0
May 9, 2024

Table of Contents

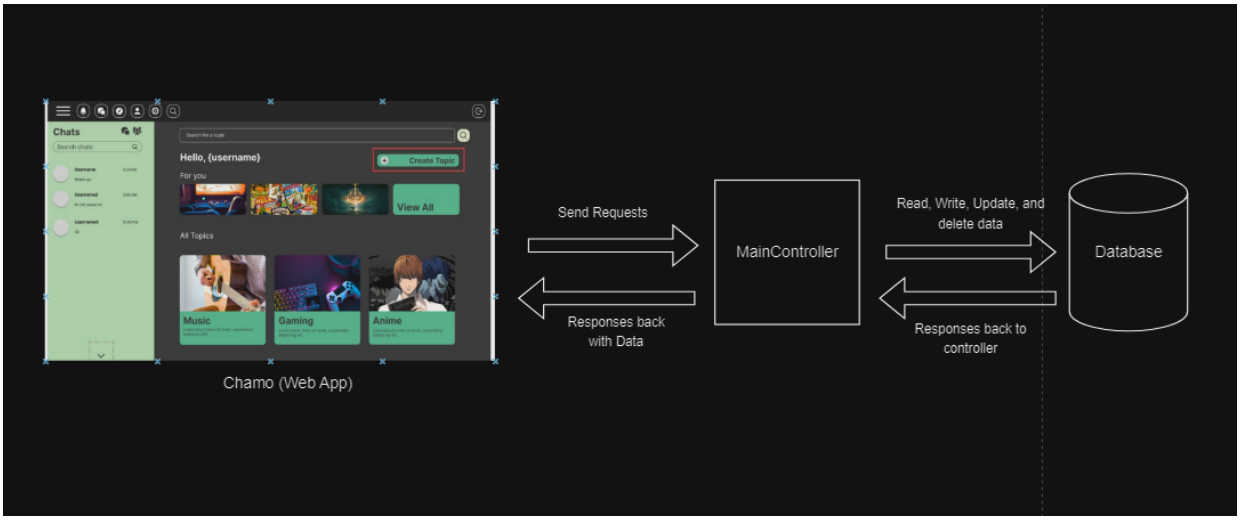
Project Overview 3
Project Component 4
Project Documentation Resources 5
Feature Specification 5
Requirement Specification Documentation 5
Design Specification Documentation 5
Project Prototyping 5
Setting Up Project Tools 6
Base Requirements 6
Additional Tools and Libraries 6
Contributing to Project Development 7
Development Process 7
Developer Support 7

Project Overview

Chamo is a web application that allows users to anonymously message one another. In comparison to similar platforms, Chamo offers a safe and welcoming platform allowing users to anonymously message one another. Ultimately, the web application aims to decrease users' judgment of other users at the beginning, and let users interact based on a certain criterion (e.g., personality test or similar interest) or no criterion (randomization), and users will have the option to connect with another user on a "deeper level" (sharing real profile).

The document provides an overview on how developers can get set up and contribute to the project.

Project Component



Web App	MainController	Database
The web app, Chamo, will act as the user's interface. Additionally, it will contain all the main functionality and features. This includes the login/signup, explore page, chat feature, settings, and many more. All information from the explore page, chat, and users will be stored in MongoDB	There will be sub controllers that will handle logic for a particular feature of the web app. But in general, there will be a main controller that's responsible for performing CRUD operations for the app.	All information such as user's profile, topics, chats, and mutual friends will be stored in the database. Chats will only be deleted if the user deletes them. There will be no time expiration date.

Project Documentation Resources

This section provides links to documentation that outline the scope and features of Chamo. Please refer to appropriate documentation below to find more information regarding a topic related to the development of Chamo.

Feature Specification

Outlines the core features of Chamo

Link: Features List Document

Requirement Specification Documentation

Outlines the stakeholder model, goal model, system vision, usage model, and functional requirements of Chamo.

Link: Requirements Specifications Document

Design Specification Documentation

Outlines the system overview, technologies used, behavior specification, and architecture specifications.

Link: Design Specifications Document

Project Prototyping

Designs of the layout and features of the project using Figma as the software tool.

Link: Prototyping Document

Setting Up Project Tools

Base Requirements

This project makes use of the following tools:

1. **Visual Studio Code:** A free IDE that is more than powerful enough to create this project. Link to the IDE: https://code.visualstudio.com/
2. **JavaScript:** This will be our main programming language for this project. It will be used for libraries like React, Express.js, and many more. Reference to their documentation: https://developer.mozilla.org/en-US/docs/Web/JavaScript
3. **ReactJS:** Our front-end library that will be responsible for creating the UI. With the use of components, states, and hooks, this will hopefully speed up development while retaining the complexity of the app. Link to the documentation: https://react.dev/reference/react
4. **NodeJS & ExpressJS:** NodeJS is a backend runtime environment that can execute JavaScript code on a system. ExpressJS is a backend framework for building RESTful APIs. Our controllers will be running on the cloud.
    a. Express.js: https://expressjs.com/en/4x/api.html
    b. Node.js: https://nodejs.org/en/docs
5. **GitHub & Git:** A version control is a must for an ambitious project like this. Each developer will choose to either use the terminal or GitHub Desktop.
    a. Git: https://git-scm.com/doc
    b. GitHub: https://github.com/

Additional Tools and Libraries

In addition, this project will make use of the following resources:

1. **Front-end:**
    a. React Router – Manages routes among pages in the app.
        i. https://reactrouter.com/en/main
    b. TailwindCSS - A CSS framework for rapidly developing the UI without dealing with CSS
        i. https://tailwindcss.com/docs/installation
    c. Axios - A HTTP client for node.js
        i. https://axios-http.com/docs/intro
2. **Back-end:**
    a. Websocket (Socket.io) - JavaScript library for real-time bi-directional communication via websockets
        i. https://socket.io/docs/v4/
3. **Database management:**
    a. Mongoose - Object Data Modeling that expresses stored data into schemas.
        i. https://mongoosejs.com/docs/
4. **Password security and management:**

   a. Bcrypt - Password hashing for secure storage of passwords
      i. https://github.com/dcodeIO/bcrypt.js/blob/master/README.md
5. **API management:**
   a. JSON Web Tokens – Maintaining secure authentication between users in exchanging information to the backend.
      i. https://jwt.io/libraries

Contributing to Project Development

Contributions to this project are made and located on Github, which can be accessed at the following Github Link.

How to run it on your own machine

Create two terminals in VSCode. Change the directory to client and server folder respectively. To run the server, type in *node index.js* in the cmd line. When done successfully, the terminal should print out "Server is running on Port ###". Next, in the client terminal, type in npm start to launch the react app. When done successfully, there should be no errors and a window of an app pops up.

Development Process

Contributions are made to the GitHub repository. Developers will each have a branch from the main codebase. Each developer works on their features in their separate branch first. Before each feature is merged into the main codebase, the feature must be reviewed by the repository administrators to ensure it is working properly. If approved, the branch will be merged; else if not, adjustments to the changes will be made until approved.

Developer Support

For more information regarding project support and details, contact any of the developers listed below.

Alex Kem: alex.kem@student.csulb.edu

Angeline Dequit: angeline.dequit@student.csulb.edu

Darren Seng: darren.seng@student.csulb.edu

Ivy Ly: ivy.ly02@student.csulb.edu

Sopheak Chim: sopheak.chim@student.csulb.edu